Effects of NRZ-M Modulation on Convolutional Codes Performance

L. Deutsch, F. Pollara, and L. Swanson
Communications Systems Research Section

Non-Return-to-Zero Mark (NRZ-M) modulation is often used to resolve data sense in suppressed carrier telemetry systems. This is because such systems are subject to half cycle slips that result in complementing the encoded data stream. The performance of coded telemetry systems with NRZ-M is sensitive to the order in which the various operations are done. This means that a system that demodulates the NRZ-M waveform and then decodes will perform differently from a system that does the decoding first. In this report, the performance of the NASA standard (7, 1/2) convolutional codes is determined for several systems using NRZ-M. In addition, several different demodulation schemes for NRZ-M are considered. It is shown that, even for the best soft-decision method examined, there is a 2.7 dB loss at a decoded bit error rate of 5×10^{-3} if the NRZ-M demodulation occurs before rather than after Viterbi decoding.

I. Introduction

Emerging coding standards for interagency projects, such as those adopted by the Consultative Committee for Space Data Systems, will include Non-Return-to-Zero Mark (NRZ-M) modulation (Ref. 1) on suppressed carrier links. This type of modulation, used to avoid serious errors in case of half-cycle carrier synchronization slips, is defined as follows.

Suppose that $\{x_n\}$ is a binary stream of bits from $\{0,1\}$. This conventional binary representation of data is sometimes called "Non-Return-To-Zero Level" or NRZ-L modulation. The NRZ-M encoder output $\{y_n\}$ is defined by

$$y_n = x_n \oplus y_{n-1}$$

where $\oplus$ denotes addition modulo two.

Decoding (i.e., translation back to the original NRZ-L binary stream) is done by noting that $x_n = y_n \oplus y_{n-1}$.

The present DSN uses residual carrier transmission exclusively, which is subject only to full-cycle slips. In the DSN case NRZ-L modulation is therefore appropriate.

If a suppressed carrier system were used with NRZ-L modulation, a half-cycle slip, which corresponds to an inversion of the data stream, could easily destroy large blocks of data. With NRZ-M the inversion would cause only isolated errors. If there is an outer block code, such as the NASA standard Reed-Solomon code, then these isolated errors are likely to be corrected by the block decoder.

This article studies the effects of NRZ-M modulation on a (7, 1/2) convolutionally coded system. (The coding and decoding for such systems is not discussed here but a good presentation may be found in Ref. 2.) In particular, two different schemes for implementing NRZ-M will be examined. The first, called Inner NRZ-M, introduces NRZ-M encoder and decoder just before and after the channel, so that the NRZ-M modulation operates on convolutionally encoded bits. The second scheme introduces encoder and decoder before the convolutional encoder and after the Viterbi decoder, respectively, and will be called Outer NRZ-M. (Outer NRZ-M is an option only in the case of transparent codes [Ref. 2] such as the NASA standard convolutional codes.)

A detailed description of these two schemes is in Sections III and IV. Their performance with various decoding methods was determined by analysis and software simulation. These are described in Sections V and VI. The conclusions of the study follow immediately in Section II.

II. Conclusions

The addition of NRZ-M outside the convolutional channel (Fig. 1) causes only a slight drop in performance on a Gaussian channel whether symbols are hard- or soft-quantized, and so is a reasonable solution to the problem of half cycle slips in the case of suppressed carrier tracking.

The other possible implementation of NRZ-M, inside the convolutional channel (Fig. 2) causes far more performance degradation, and should be avoided.

Numerical results are shown and explained in Section VI.

III. NRZ-M Modulation

This type of modulation, widely used to avoid phase ambiguities of half cycles or multiples, is also and more properly referred to as "differential encoding" (Ref. 1).

Suppose that $\{x_n\}$ is the binary data stream to be encoded. Then the encoder output $\{y_n\}$ is defined as in Section I by:

$$y_n = x_n \oplus y_{n-1}$$

If we consider the sequence $\{x_n\}$ to be bipolar data (i.e., each x_n is taken from the set $\{-1,+1\}$ rather than $\{0,1\}$), then we may rewrite this expression as

$$y_n = -x_n y_{n-1}$$

This notation is used for the remainder of this article.

Let the received sequence be $\{\hat{y}_n\}$, where $\hat{y}_n$ is the result of passing y_n through a binary symmetric channel. In order to restore the original data, we must appropriately decode the received sequence $\{\hat{y}_n\}$. Let the NRZ-M decoder output be $\{\hat{x}_n\}$, where

$$\hat{x}_n = -\hat{y}_n \hat{y}_{n-1}$$

Then, if there are no errors in the received sequence, i.e., $\hat{y}_n = y_n$ for each n, we have

$$\begin{aligned}\hat{x}_n &= -\hat{y}_n \hat{y}_{n-1} \\ &= x_n y_{n-1} y_{n-1} \\ &= x_n\end{aligned}$$

as required. Note that if both y_{n-1} and y_n are complemented, we still have $\hat{x}_n = x_n$. This invariance under signal inversion is the property that makes NRZ-M an attractive algorithm for suppressed carrier systems.

The encoder/decoder algorithm just described can be directly used to implement the Outer NRZ-M scheme.

The same is true for the Inner NRZ-M, if the received data is hard quantized. Otherwise, for a soft quantized receiver, we need to modify the NRZ-M decoder in some way.

With NRZ-L data, alternate symbols from the convolutional encoder are usually inverted. This increases received symbol transition probability, resulting in improved symbol synchronization (Ref. 3). In the case of Outer NRZ-M encoding, such a system would increase symbol transitions in the same way, with no change in the results of this report. In the case of Inner NRZ-M encoding, alternate symbol inversion schemes do not guard against long strings without transitions.

IV. Soft-Decision NRZ-M Decoding

The implementation of Inner NRZ-M encoding/decoding requires either a new Viterbi decoder that operates on NRZ-M encoded symbols, or an algorithm for the NRZ-M decoder to send soft quantized symbols to the Viterbi decoder. Conventional NRZ-M decoding described in Section III produces hard-quantized symbols as an output, but use of these in the Viterbi decoder would cause a substantial loss in Viterbi decoder performance. An optimal system based on channel probabilities is computationally cumbersome.

One system which has been suggested is

$$\begin{aligned}\hat{x}_n &= -sgn(\hat{y}_n \hat{y}_{n-1})|\hat{y}_n| \\ &= -(\hat{y}_n \hat{y}_{n-1})/|\hat{y}_{n-1}| \qquad (1)\end{aligned}$$

This algorithm is the one used in the Symbol Synchronizer Assembly (SSA) in the DSN, according to the Symbol Synchronizer Assembly Technical Manual, Operations and Maintenance, and it agrees in sign with hard-quantized NRZ-M decoding.

Another system which agrees in sign with hard-quantized NRZ-M decoding is

$$\hat{x}_n = -\hat{y}_n \hat{y}_{n-1} \tag{2}$$

This second system, which appears more natural to us, performed better in the Viterbi decoder simulations (see Section VI).

V. A Theoretical Comparison of the Inner and Outer Schemes

In the case that received information over the channel is assumed to be hard quantized, a mathematical analysis of inner and outer NRZ-M systems at high SNRs is possible. This analysis consists of considering channel error sequences that produce Viterbi decoder bit errors. The probability of decoded bit error is then approximated by a sum over these sequences of their probability of occurrence multiplied by the number of bit errors caused by each one of them.

Consider the Inner NRZ-M scheme. Also, assume that the sequence that is sent is the all zero sequence. Since both Viterbi decoding and NRZ-M decoding are linear operations, there is no loss of generality in this assumption as long as the system is synchronized. Since we are only considering hard quantization in this analysis, the channel may be modeled as a binary symmetric channel (BSC) with transition probability p.

In the NRZ-L system, the Viterbi decoder (either JPL [Ref. 4] or Goddard conventions) is capable of correcting any error pattern of Hamming weight less than five (Ref. 4). A computer search was performed to determine sequences that cause bit errors when they are passed first through an NRZ-M decoder and then the Viterbi decoder. Since a weight two channel error pattern can only become at most a weight four pattern after NRZ-M decoding, all such patterns are corrected by the inner NRZ-M system. All weight three channel error patterns of length up to 16 symbols were checked by using a software Viterbi decoder. It was found that there are weight three sequences that produce bit errors in both the JPL and Goddard decoders if they were first NRZ-M decoded. In fact, there were sequences that produce eight and five bit errors, respectively, for these two codes. A complete listing of these sequences is shown in Table 1.

At very high SNRs, the probability of a weight three channel error sequence occurring at any time is approximately p^3. Suppose the weight three error sequences are $\{v_i\}$ (i=1,2, . . ., n) and that the number of decoded bit errors that each produces is $b(v_i)$. Then for very high SNRs, the probability of decoded bit error is given by

$$P_{\text{bit}} = \sum_{i=1}^{n} b(v_i) p^3 (1-p)^{13}$$

The channel transition probability p is given by

$$p = Q(\sqrt{2E_s/N_o})$$
$$= Q(\sqrt{E_b/N_o})$$

where E_s is the energy in a channel symbol and Q is the Gaussian error function

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp(-v^2/2)dv$$

For large x we may use the approximation (Ref. 5)

$$Q(x) = \frac{1}{\sqrt{2\pi}\, x} \exp(-x^2/2)$$

Let $\alpha = E_b/N_o$. Then, for large α we obtain

$$P_{\text{bit}} = \left[\sum_{i=1}^{n} b(v_i)\right] (1/2\pi\alpha)^{3/2} \exp(-3\alpha/2)$$

As α becomes large, this probability behaves like its exponential term. Hence, asymptotically,

$$P_{\text{bit}} = \exp(-3\alpha/2) \tag{3}$$

for both the JPL and Goddard codes. In comparison, these codes, without NRZ-M on the channel, behave asymptotically like

$$P_{\text{bit}} = \exp(-5\alpha/2) \tag{4}$$

Suppose that the bit SNR required to achieve a certain error rate for the conventional system is β and that the SNR required to get the same performance in the Inner NRZ-M system is α. Then by Eqs. (3) and (4),

$$\exp(-3\alpha/2) = \exp(-5\beta/2)$$

or

$$\alpha = (5/3)\beta$$

This means that the inner NRZ-M scheme should behave about 2.2 dB worse than the NRZ-L system in the limit as SNR becomes infinite. For "reasonable" signal-to-noise ratios, the difference is even greater because of the terms other than the exponential terms. In particular, to obtain bit error probability 0.005, $\alpha = 2.04\beta$, about a 3 dB loss.

If NRZ-M decoding is added after the Viterbi decoder, then the bit errors produced by the decoder are at most doubled. This constant factor is not significant in asymptotic behavior. Therefore the Inner NRZ-M scheme should also behave 2.2 dB worse than Outer NRZ-M for very high SNRs.

VI. Simulation Results

In order to determine the performance of various telemetry system configurations involving NRZ-M and convolutional coding, software simulations were carried out. The software modules included a convolutional encoder, Viterbi decoder, NRZ-M encoder, NRZ-M decoders for each of the NRZ-M decoding algorithms described above, and a Gaussian channel simulator. These were written in C-Language and run on a VAX 11/750 computer under the UNIX operating system.

The configuration used for the simulation is shown in Fig. 3. The "generator" output is a stream of 0's and 1's. The "coder" is a convolutional encoder implementing one of the standard NASA (7,1/2) codes. The "display" routine compares the delayed information bits with the decoded bits and generates the statistics. In addition there are NRZ-M coders and decoders which may be inserted in various places.

Figure 4 shows all configurations used in the simulation. Results are shown in Figs. 5 and 6, with decoded bit error rate as a function of the bit signal-to-noise ratio, E_b/N_o, for each scheme. The simulations assume perfect carrier and subcarrier tracking and perfect Viterbi decoder node synchronization. The Viterbi decoder path memory length was 32 bits.

The probability of error P_E is measured on a random sequence of N = 1,000,000 bits or until the 95% confidence interval C, defined as

$$C = 1.96\sqrt{P_E(1-P_E)/N}$$

is less than 5% of P_E. Therefore the true P_E is inside the range $P_E \pm C$ with 95% confidence. Table 2 shows a comparison of the various schemes for bit error probabilities of 10^{-2}, 5×10^{-3}, and 10^{-3}.

As expected, the Outer NRZ-M scheme outperforms the Inner NRZ-M scheme, for both the JPL and Goddard codes, the latter being slightly better.

References

1. Lindsey, W. C., and Simon, M. K., *Telecommunication Systems Engineering*, Prentice-Hall Inc., 1973.
2. McEliece, R. J., *The Theory of Information and Coding*, Addison-Wesley, 1977.
3. Simon, M. K., and Smith, J. G., "Alternate Symbol Inversion for Improved Symbol Synchronization in Convolutionally Coded Systems," *IEEE Trans. Comm.*, Vol. COM-28, February, 1980, pp. 228-237.
4. Yuen, J. H., *Deep Space Telecommunications Systems Engineering*, Plenum, New York, 1983.
5. Wozencraft, J. M., and Jacobs, I. M., *Principles of Communication Engineering*, Wiley, 1965.

Table 1. A list of Viterbi-decoded bit error sequences caused by the inner NRZ-M system as a result of weight three channel symbol error sequences. Hard quantization is assumed.

JPL (7, 1/2) code		Goddard (7, 1/2) code	
Viterbi-Decoded Bit Error Sequence	Number of Errors	Viterbi-Decoded Bit Error Sequence	Number of Errors
1001000000000000	2	1001000000000000	2
1000000000000000	1	1000000000000000	1
1011000000000000	3	1100100000000000	3
1100000000000000	2	1100000000000000	2
1100100000000000	3	1011000000000000	3
1010100000000000	3	1110010000000000	4
1010000000000000	2	1111001000000000	5
1111001000000000	5	1110000000000000	3
1111110000000000	6	1111000000000000	4
1011011000000000	5	1101100000000000	4
1111000000000000	4	1010010000000000	3
1100111111000000	8	1010000000000000	2
		1010100100000000	4

Table 2. E_b/N_0 required to achieve a given P_E for different schemes

P_E	Goddard Code				JPL Code			
	NRZ-L Soft	Outer NRZ-M Soft	Inner NRZ-M Soft[a]	Inner NRZ-M Soft	NRZ-L Soft	Outer NRZ-M Soft	Inner NRZ-M Soft[a]	Inner NRZ-M Soft
10^{-1}	0.5	0.65	3.1	4.0	0.45	0.55	3.1	4.0
10^{-2}	1.8	1.9	4.55	5.65	1.85	1.95	4.55	5.6
5×10^{-3}	2.1	2.2	4.85	5.95	2.15	2.25	4.85	5.95

[a]Method of Eq. (2)

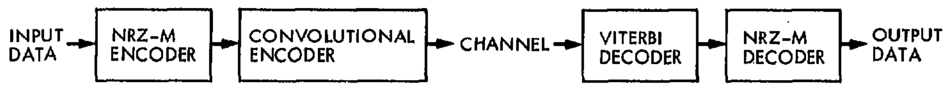

Fig. 1. Outer NRZ-M coding system

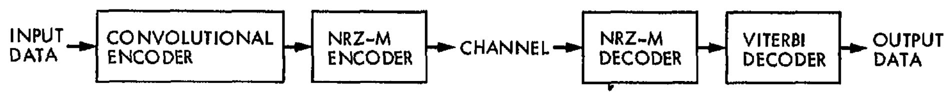

Fig. 2. Inner NRZ-M coding system

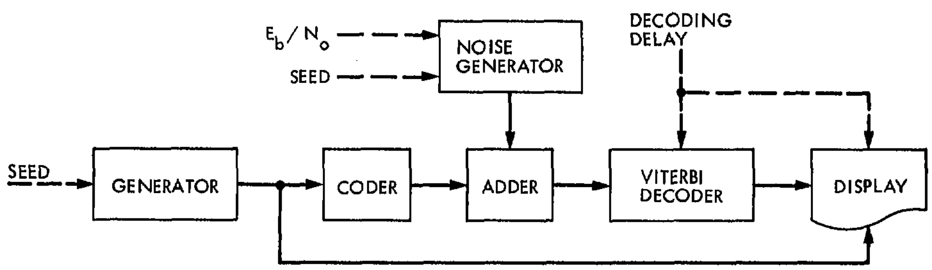

Fig. 3. Block diagram for simulation of NRZ-M with convolutional codes

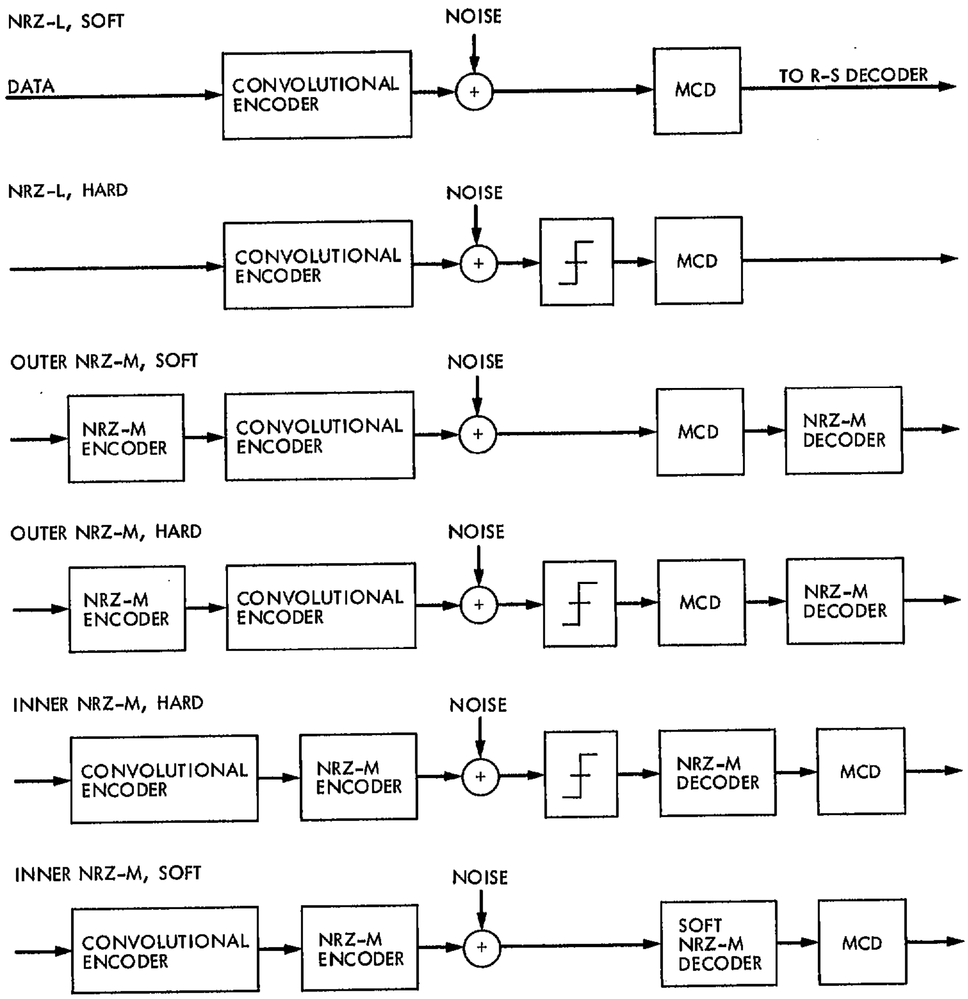

Fig. 4. Configurations used in simulation

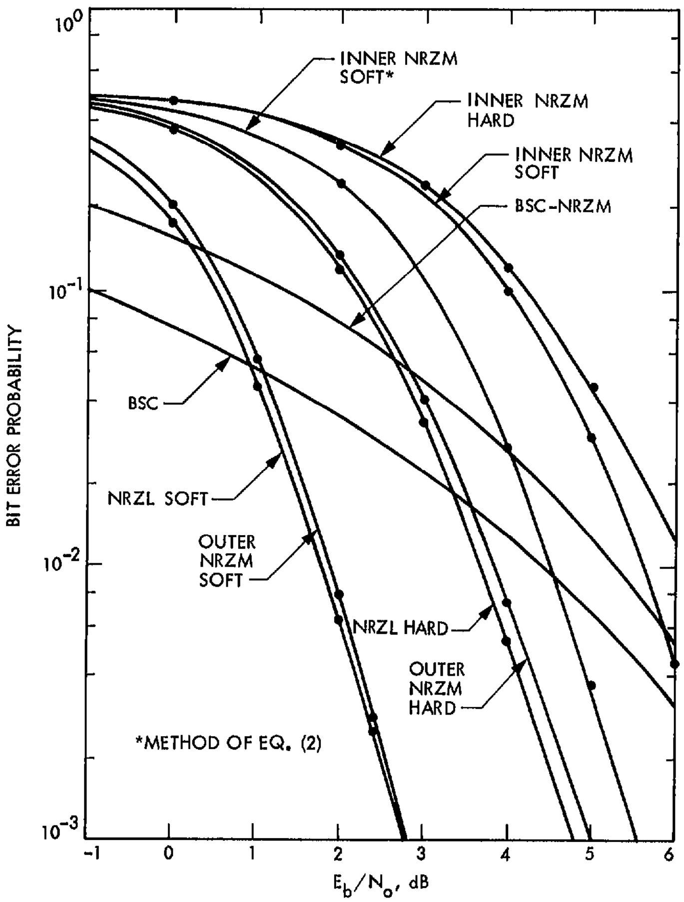

Fig. 5. Performance of Goddard (7, 1/2) convolutional code under various NRZ scenarios

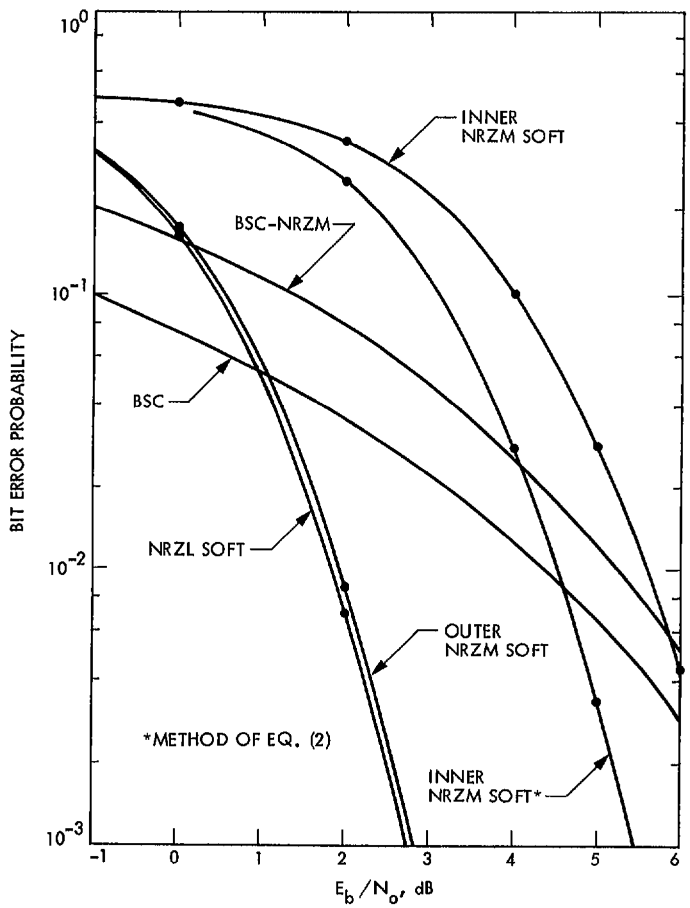

Fig. 6. Performance of JPL (7, 1/2) convolutional code under various NRZ scenarios